

TERCERA PRÀCTICA DE PROGRAMACIÓ

Gestor d'associacions a l'ETSE

Presentació del problema

Cada vegada estan apareixent noves associacions d'estudiants a l'ETSE. Per això volem implementar una aplicació per facilitar la seva gestió. *L'enunciat és un supòsit amb finalitats docents, per tant, els requeriments que es plantegen no tenen per què coincidir amb la realitat.*

Requeriments d'informació

El sistema ha de gestionar les dades de les diferents associacions, portar un control dels membres de cada associació i les accions que van duen a terme. A continuació detallem més la informació.

Associació

De cadascuna en guardarem el nom (que l'identifica) i correu electrònic de contacte. També es guardarà informació de les titulacions, la majoria de l'ETSE, que cursen els seus membres. Identificarem les titulacions amb les seves sigles (GEB, GEI, GESST, BioGEI, DG GEB-GESST, extern-ETSE). Tindrem una llista d'associacions.

Les associacions estaran formades per un conjunt de membres. Del conjunt de membres actius hi haurà tres persones que ocuparan els càrrecs de president, secretari i tresorer (càrrecs que només poden estar ocupats per alumnes).

Membres: alumnes i professors

Les associacions estan formades per persones, en la major part dels casos per alumnes però també hi ha alguns professors/es que hi col·laboren activament. Qualsevol persona pot demanar pertànyer a una associació. Si s'accepta, es dona d'alta en una data. La persona pot demanar deixar de formar part de l'associació i llavors es guardarà en quina data s'ha donat de baixa. No s'esborraran les dades dels membres que es donen de baixa, es registrarà la seva data de baixa. Un alumne no es donarà de baixa i posteriorment de nou d'alta (és a dir, no entra i surt de l'associació diverses vegades). Els alumnes que acaben l'ensenyament poden continuar a l'associació però ha de constar que l'alumne ja està graduat.

Programació Curs 2024-2025

De cada membre es guardarà un alies (que l'identifica) i la seva adreça de correu electrònic institucional. En el cas dels alumnes es guardarà també l'ensenyament on estan matriculats (via les seves sigles) i el número d'anys que porten a l'ETSE. En el cas dels professors es guardarà el nom del departament al que pertanyen (serà DEIM o DEEEA) i el número del seu despatx.

Un membre es pot apuntar a diferents associacions i estar actiu simultàniament en varies, però com a molt de 3 associacions diferents al llarg de tot el temps.

Tindrem una llista de membres.

Accions que duen a terme: demostracions i xerrades de divulgació

Totes les accions que organitza una associació tenen un codi que les identifica, que es genera automàticament a partir de les 3 primeres lletres del nom de l'associació, i seguides d'un número (començant per 100). De totes les accions guardarem també un títol. Les accions estan organitzades per una o més associacions. Tenen un responsable de l'acció que és membre d'alguna de les associacions que organitza.

Les demostracions són accions preparades per oferir en diferents jornades, portes obertes, setmana de la ciència, ... De cada demostració guardarem la data en què es va dissenyar i si encara és una acció que es considera vàlida i per tant es continua oferint. Volem comptabilitzar quantes vegades s'ha ofert aquesta demostració, guardarem aquesta informació i també el cost que va suposar crear els materials per a la demostració.

Les xerrades són accions que es preparen per una data concreta. No es repeteix mai cap xerrada perquè sempre s'adapta a les necessitats del públic. En aquesta acció guardarem la data en què s'ha fet i les dades dels membres que l'han impartit (podem suposar un màxim prefixat, inicialment a 3). També guardarem quants assistents ha tingut la xerrada i les valoracions que aquests han fet (les valoracions estaran en l'escala [0-10]).

Tindrem una llista d'accions.

Requeriments no funcionals del disseny

Tingueu en compte que, a més de les classes corresponents als elements que s'han acabat de descriure, heu d'implementar les classes de les **l·listes** que els emmagatzemen. Es recomana que en cada llista els elements estiguin ordenats per l'atribut més adient segons les cerques que es demanen (veure més avall). ATENCIÓ: No es poden usar les classes de Java per guardar les l·listes (com ArrayList, o altres). Recordeu que els mètodes de la llista no han de retornar sempre String, sinó que han de retornar les dades del tipus que correspongui, i serà la part de visualització la que s'ocupi de mostrar-les com calgui.

Les classes que emmagatzemen la informació de l'aplicació es faran persistents mitjançant **fitxers**. A cada nova execució del programa, es carregarà automàticament tota la informació dels fitxers i es crearan i inicialitzaran els objectes de les classes corresponents. Durant l'execució de l'aplicació, les

dades s'aniran modificant sobre els objectes de memòria com a resultat de les funcionalitats demanades pels usuaris. Quan es vulgui tancar l'aplicació, es demanarà a l'usuari si vol guardar tota la informació que hi ha als objectes de memòria als fitxers, sobreescrivint el contingut que tenien (aquesta opció està pensada per a facilitar/actualitzar els jocs de proves).

Les dades de les associacions es guardaran en un **fitxer serialitzat**. La resta d'informació es guardarà en **fitxers de text**. Tots els fitxers de text han de seguir un format on les dades de cada instància han d'estar en una línia amb els diferents atributs separats pel caràcter ';'. A la documentació final s'ha d'explicar el format dels fitxers utilitzats, posant algun exemple. També cal incloure els fitxers de prova al projecte que s'entregarà. Els fitxers de prova han de ser suficientment complets per poder comprovar les diferents funcionalitats demanades. També hi ha d'haver un nombre raonable de dades que garanteixin que heu provat suficientment la vostra aplicació.

S'haurà d'afegir el **control d'excepcions**. S'han de tenir en compte les excepcions predefinides en Java que es puguin donar. També s'ha de definir alguna excepció pròpia que sigui coherent amb els requeriments de la pràctica. La **documentació** també ha d'explicar detalladament tots els errors que es controlen, on i amb quina excepció (predefinida o pròpia).

La interfície gràfica que es demana més avall s'ha d'implementar en un package a banda, separada de les classes de dades. Aquesta interfície s'implementarà usant les classes SWING i es programarà des de zero (no es pot usar cap eina d'implementació d'interfícies automàtica).

No es poden usar eines de generació de codi automàtic, en cap de les parts de la pràctica.

Requeriments funcionals

Es demana realitzar dues aplicacions, una amb entrada i sortida per consola, i una altra on l'entrada i sortida es farà mitjançant interfície gràfica. A continuació es detallen les funcionalitats que ha d'oferir cadascuna d'elles. **Important:** totes les aplicacions treballaran sobre la mateixa informació, és a dir, les mateixes dades carregades dels mateixos fitxers.

Aplicació amb entrada/sortida per consola

L'aplicació carrega la informació dels fitxers i manega l'entrada/sortida per la consola. Abans de mostrar el menú amb les funcionalitats es llegiran les dades dels fitxers i s'inicialitzarà les estructures llistes amb tota la informació.

Tot i que si fos una aplicació real necessitaríem més opcions al programa, donat que això és només un exercici docent, us demanem només implementar les que s'indiquen a continuació.

Les opcions de menú han de ser:

1. Mostrar les dades de la llista d'associacions.
2. Mostrar les dades de la llista de membres que formen part d'una associació, afegint filtre per a professors, alumnes o ambdós.
3. Mostrar les dades de la llista de membres actius, que formen part de qualsevol associació, afegint filtre per a professors, alumnes o ambdós.
4. Mostrar les dades de la llista d'accions, afegint filtre o no per tipus d'acció.

5. Obtenir i mostrar la llista d'accions que ofereix una associació concreta.
6. Obtenir i mostrar la llista de les xerrades que es duen a terme en una franja de dates indicada per teclat.
7. Afegir una nova associació.
8. Alta d'un membre a una associació. Es pot donar el cas que el membre sigui nou, i s'haurà d'introduir tota la informació del membre, o que el membre ja participi en una altra associació i en aquest cas serà afegir la relació corresponent.
9. Afegir una nova xerrada.
10. Afegir una nova demostració.
11. Consultar i mostrar les dades de les demostracions que es consideren no actives. Calcular el cost econòmic total que va suposar preparar totes aquestes demostracions.
12. Calcular la persona més activa, és a dir, la que participa en més associacions. En cas d'empat es considera la que té més antiguitat (en qualsevol associació). Si encara hi ha empat, s'escull qualsevol de les persones que compleixen els requisits.
13. Consultar i mostrar les dades de les xerrades que ha tingut més d'un cert nombre indicat d'assistents.
14. Valorar una xerrada per part d'un assistent.
15. Consultar i mostrar la xerrada que està millor valorada (que serà la que té la mitjana de valoracions més alta), en cas d'empat en la nota, considerar la que ha tingut més valoracions i en cas d'empat agafar qualsevol.
16. Mostrar les dades de les xerrades que farà una persona concreta.
17. Donar de baixa les demostracions que no estiguin actives i que es van dissenyar abans d'una certa data.
18. Sortir de l'aplicació.

Aplicació amb entrada/sortida gràfica

L'aplicació carrega la mateixa informació dels fitxers que es fa servir en la versió per consola. Aquesta aplicació té un conjunt molt més reduït de funcionalitats que ofereix mitjançant una interfície gràfica. Cal implementar-ho amb les eines del paquet SWING, sense usar cap generador de codi automàtic. Podeu fer ús dels diferents elements gràfics presentats a l'assignatura per compondre la finestra. Es valorarà la facilitat d'ús de la interfície.

La funcionalitat des de la interfície serà mostrar gràficament el conjunt de demostracions que estan actives. Aquesta informació s'ha de poder filtrar per associacions, és a dir, podem voler mostrar totes les demostracions actives o només les demostracions actives ofertes per algunes associacions seleccionades. De cada demostració que es visualitza hem de poder consultar tot el seu detall.

Instruccions per al desenvolupament de la pràctica

- **Aquesta pràctica es treballa en equips de quatre persones.** A l'espai moodle compartit entre *Programació i PSI* teniu una activitat per apuntar-vos a un dels equips. Treballar en equip vol dir repartir-se les tasques necessàries per resoldre tot el treball, no vol dir fer-ho tot entre tots. Així doncs recomanem fer primer una lectura global de la pràctica, analitzar el conjunt de classes que són necessàries i repartir-vos la feina. Cada persona és responsable del bon disseny i funcionament de la seva part. Cada equip tindreu assignat un o dos supervisors de 4rt de GEI. Ells us ajudaran a fer el disseny del programa, a repartir la feina i a treballar de forma coordinada. Cal seguir les seves indicacions però ells no programen res, ni resolen dubtes tècnics. Consulteu a les professores de Programació en cas de dubtes de com implementar alguna part.
- **Cada equip tindrà assignat un/a professor/a de laboratori.** Ell/a revisarà la distribució de la feina que us feu entre els membres de l'equip i el disseny de classes inicial. Comuniqueu al professorat també el més aviat possible si hi ha algun problema amb l'equip (baixes de membres, desajustos en la planificació, dificultats amb el supervisor/a, etc). Per poder seguir la comunicació entre els membres de l'equip i la professora assignada hi ha **el fòrum d'equip**.
- La comunicació bàsica entre els membres de l'equip s'ha de fer via el **fòrum d'equip** que tindreu al campus virtual en l'espai compartit. Valorarem el vostre treball continu sobre el contingut d'aquest fòrum. Podeu utilitzar altres eines de comunicació, però no seran visibles als professors i per tant no es podran tenir en compte en l'avaluació.
- La comunicació bàsica que s'espera són les actes de les reunions de l'equip. Cal que les consulteu per comprovar els acords presos i cal que compliu el que heu acordat en cada una.
- A més, per a validar el treball i assegurar que heu plantejat un bon disseny de classes i esteu abordant el problema de forma correcta us **proposem una entrega voluntària el dia 9 de desembre de 2024**. L'entrega ha de servir per a validar a grans trets que l'estructura de classes, la forma d'utilitzar-les, ... és correcta. En aquesta data no es demana tot el codi, sobretot per a la part d'interfície gràfica no s'espera res, encara. Malgrat ser una entrega voluntària es gratificarà amb fins a 1 punt extra a la nota de la pràctica que s'avaluarà al gener. El valor de la nota extra dependrà de que hi hagi un mínim de codi fet en aquesta primera entrega.
- És molt important la validació de cada classe per separat, és a dir, fer un seguit de proves unitàries per classe. Els codis per a valorar aquestes classes s'han d'incloure en l'entrega voluntària posant-los en un Package a banda.
- En l'entrega final cal lliurar un informe final. A l'assignatura, avaluarem la competència C4 (escriptura en català o castellà). L'informe ha de seguir el format i contingut indicats a les instruccions que trobareu al campus virtual.
- El termini de lliurament de la pràctica, en primera convocatòria, és el **7 de gener de 2025**.
- A criteri del professor revisor de la pràctica hi haurà una **entrevista individual** sobre la pràctica que pot influir en la nota final. L'entrevista és individual però tots els membres de l'equip l'han de fer en la mateixa franja de temps.

- Heu de crear el *Java Project* amb el vostre número d'equip, **numEquip_nomPractica**. Recordeu estructurar el codi en diferents packages, com a mínim, un per a l'aplicació de consola, un altre per a l'aplicació amb interfície gràfica i un altre per a les classes que guarden les dades.
- Recordeu que totes les classes han d'estar ben documentades i utilitzar els comentaris JavaDoc.
- Lliurareu els codi fonts i l'informe en pdf. Tota aquesta informació es guardarà en un fitxer ZIP i es penjarà a la tasca corresponent del campus virtual. El nom del fitxer serà **numEquip_nomPractica.zip**

Criteris d'avaluació

- Referents a la implementació:
 - Correctesa en la definició de les classes (tipus dels atributs, modificadors, herència, ...).
 - Correctesa en la definició i implementació dels mètodes (que rebin els paràmetres adients i facin correctament el que es demana, de la forma més senzilla i òptima possible).
 - Manegament correcte de les llistes (recorregut/cerca).
 - Cal manegar les excepcions predefinides de Java i alguna de pròpia.
 - Lectura/escriptura de fitxer de forma senzilla i eficient.
 - El codi que s'implementi ha de ser clar: ha d'estar tabulat, ha d'estar comentat adequadament (cada mètode ha de tenir una capçalera indicant què fa, i quins paràmetres rep) i els noms que useu pels atributs, mètodes, etc han de ser representatius.
 - Ús de packages per separar el codi.
 - Ús correcte de la classe SWING per fer la interfície gràfica.
- Referents a la competència B8/CT4 i CT5:
 - Es valorarà la redacció, estil, correctesa de l'informe de la pràctica.
 - Es valorarà la distribució de tasques i treball realitzat de forma individual i de forma conjunta.
- Referents al disseny i joc de proves i al treball personal de cada membre:
 - Es valorarà el contingut de l'informe i els jocs de prova entregats.
- Altres consideracions:
 - No s'acceptaran pràctiques que no funcionin o no facin el que demana l'enunciat.
 - Reviseu que el codi compili sense advertiments.
 - En cas de detectar-se dues o més pràctiques iguals o molt similars es posarà una nota de 0 a tots els alumnes implicats.